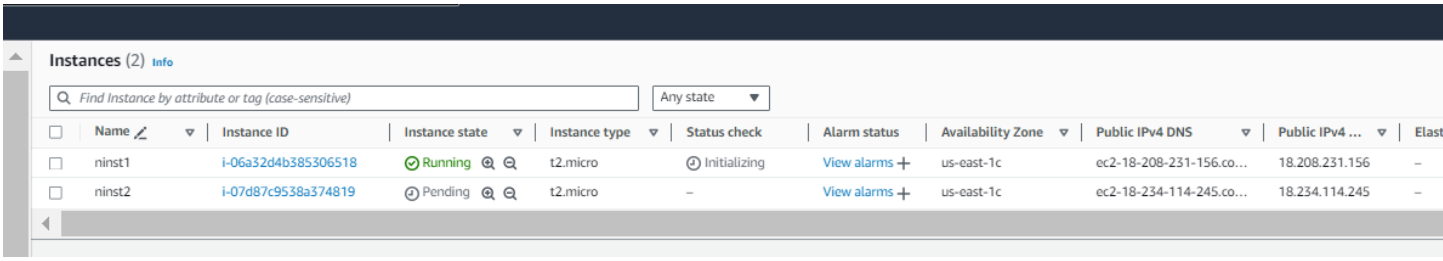
# 1)Create Network Load Balancer.

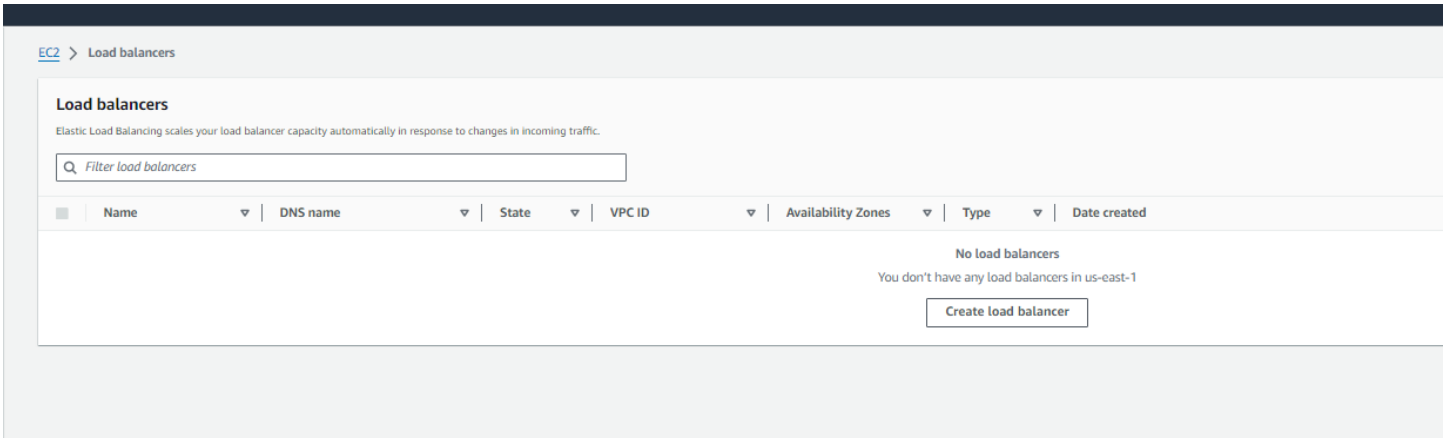
Step I : Create two instances in EC2 using below scripts in advanced settings.

```
#!/bin/bash
sudo -i
yum install httpd -y
systemctl start httpd
systemctl enable httpd
echo "this is ROOT $HOSTNAME" > /var/www/html/index.html
```

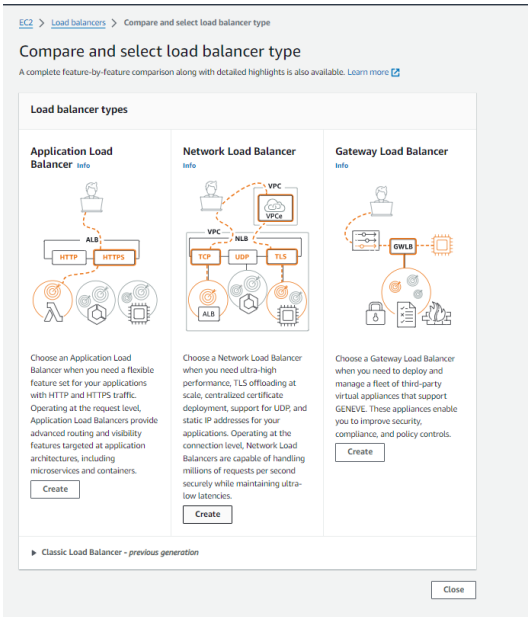
```
#!/bin/bash
sudo -i
yum install httpd -y
systemctl start httpd
systemctl enable httpd
echo "this is my HOME $HOSTNAME" > /var/www/html/index.html
```



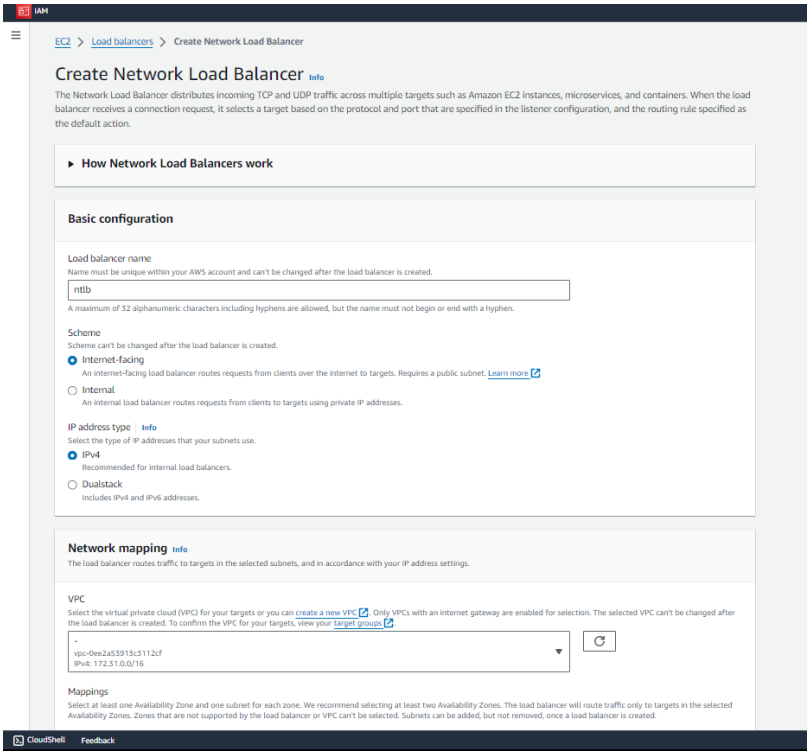

Step II : In Load Balancing, click  load Balancer and then click the create Load Balancer.

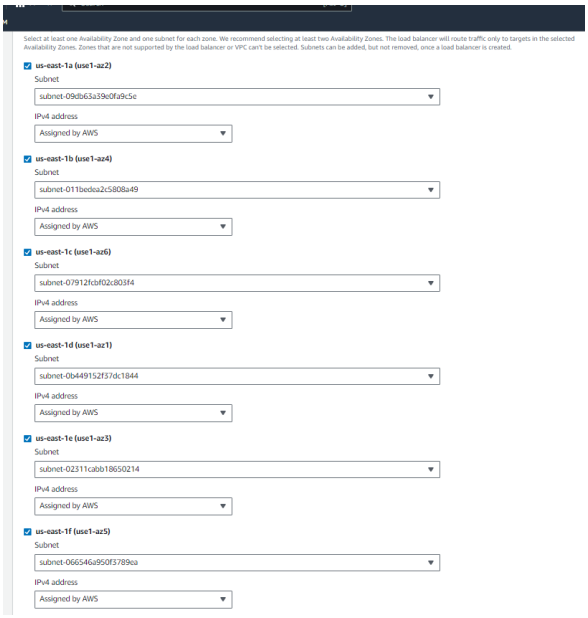


Step III : then Select the network Load Balancer.

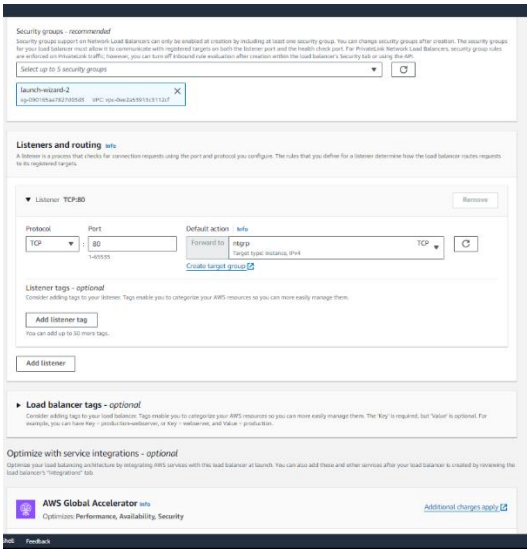


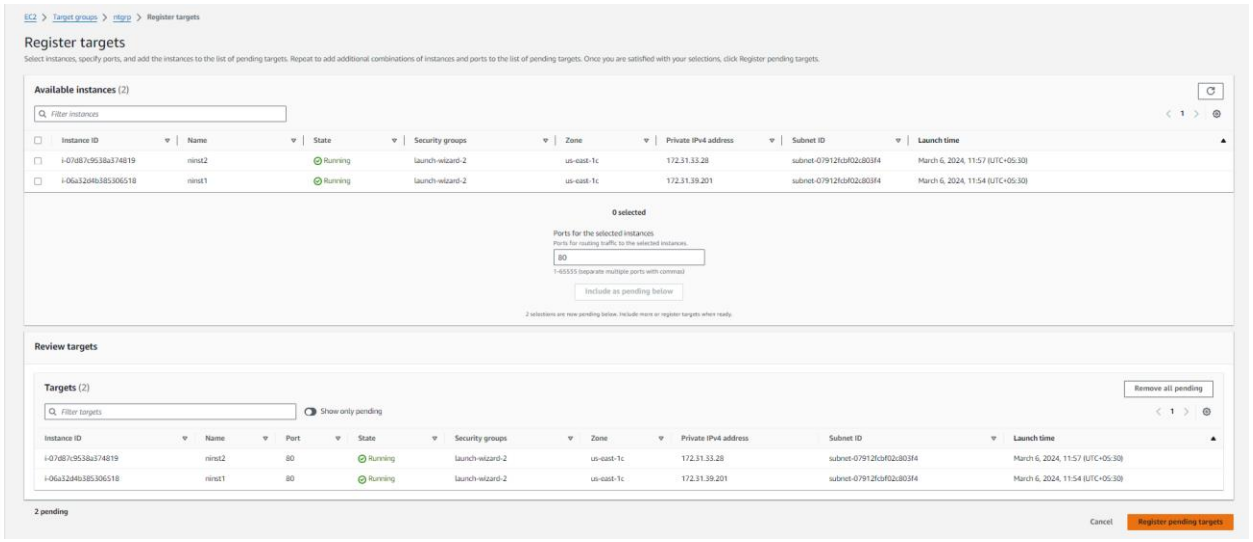

Step IV : give the name,select scheme,Ip address type, select all  zones .



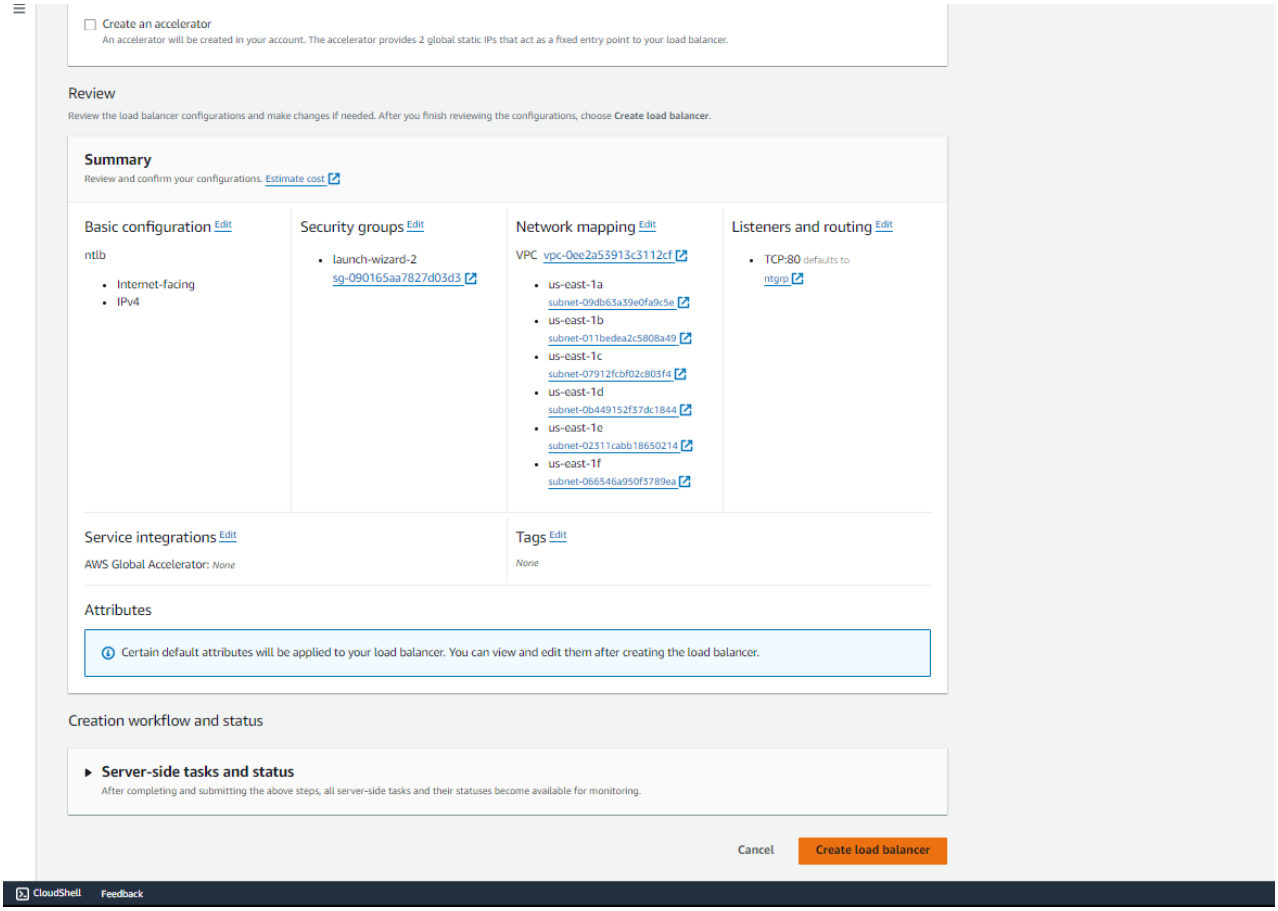



Step V : select security group, here in default action click to create target group. create the target group add the instances which we created before and registed it .Add it in Default Action.

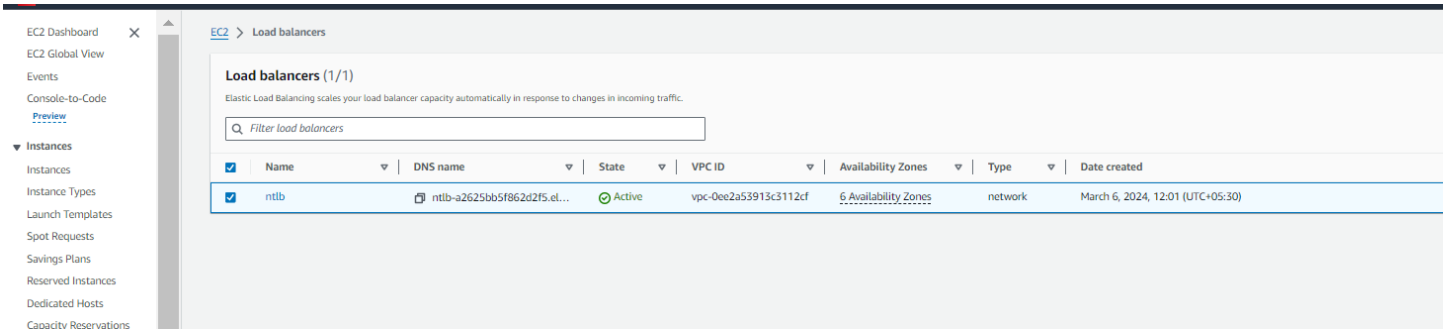


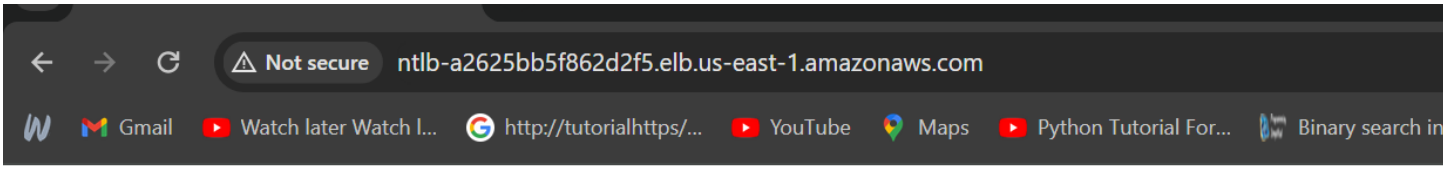



Step VI : review the summary and click on create Load Balancer.



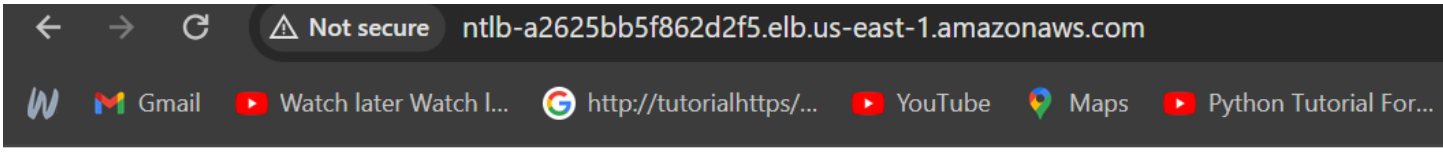
Step VII : here, load balancer is created. Copied the DNS Name and paste in other browser.




this is ROOT ip-172-31-33-28.ec2.internal

Step VIII : network load balancer directs the traffic unevenly on reload, it not used round-robin method IP get changes only at the high request occurs.



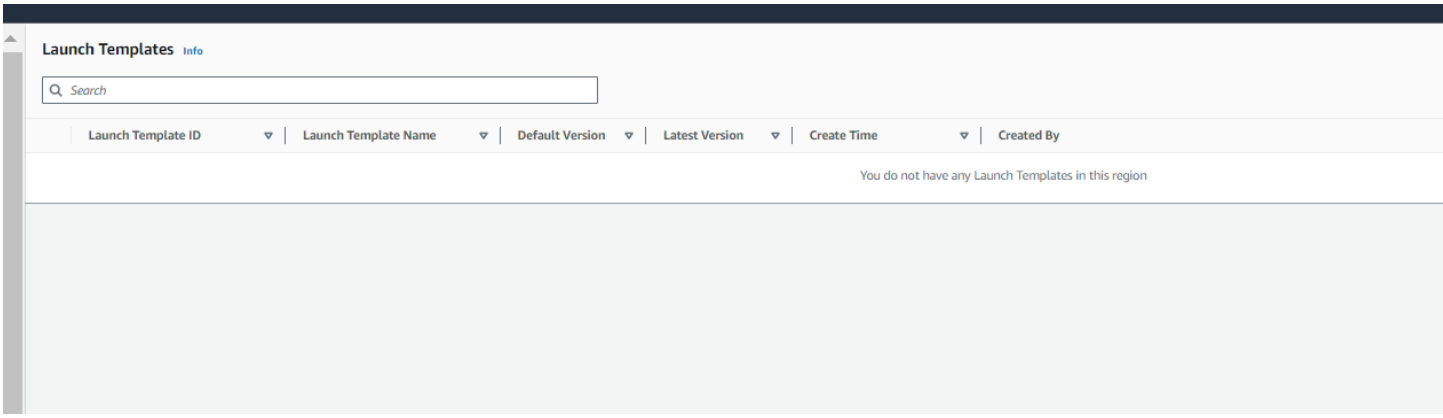

this is my HOME ip-172-31-39-201.ec2.internal

## 2)AUTO SCALING

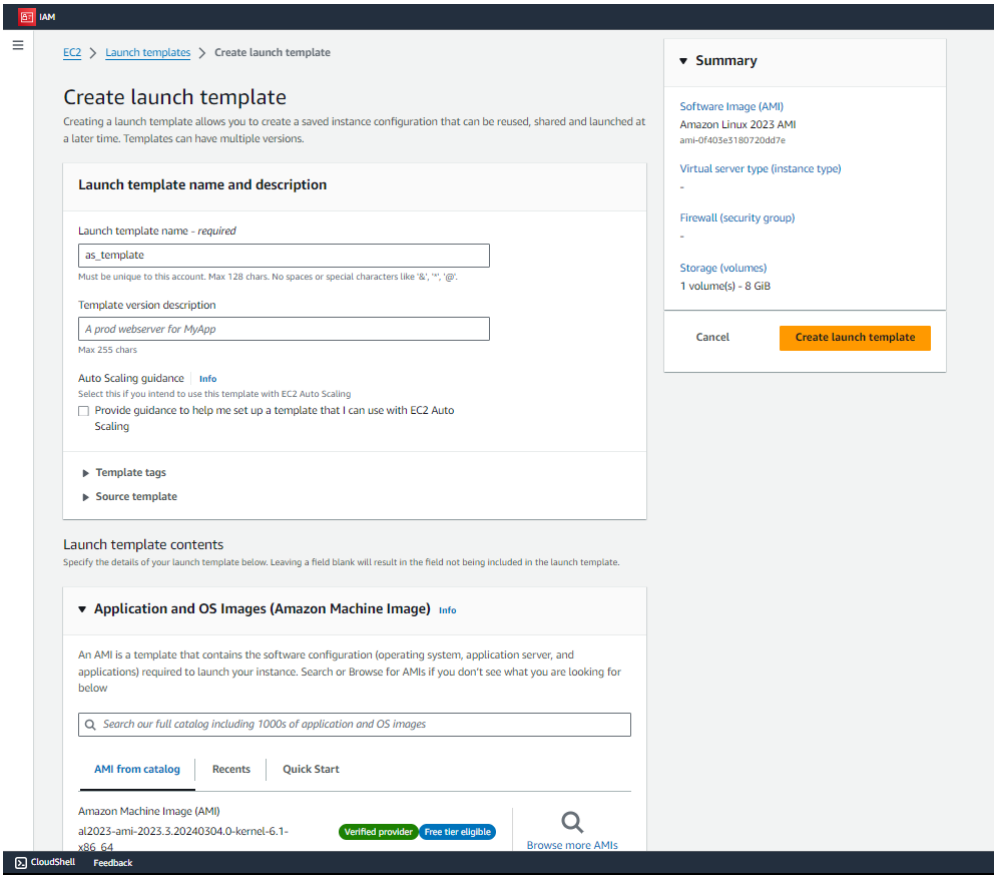
It helps you automatically adjust the number of Ec2 instances in your application based on demands.

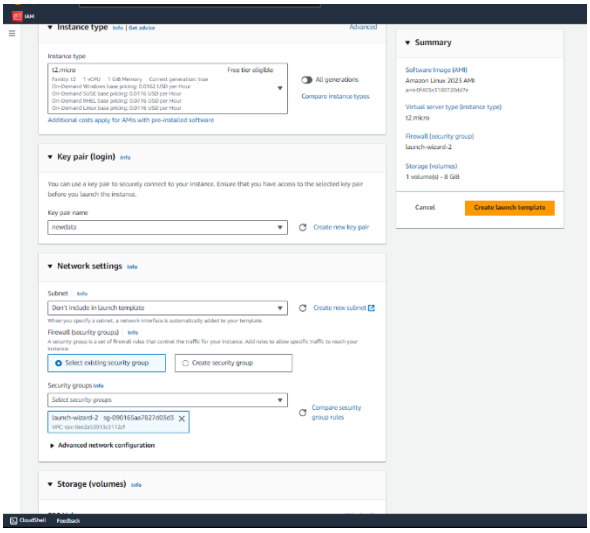
### 1)Create the Template.

Step I : go to Ec2 ,In Instances click on Launch Template then click on create  launch template.

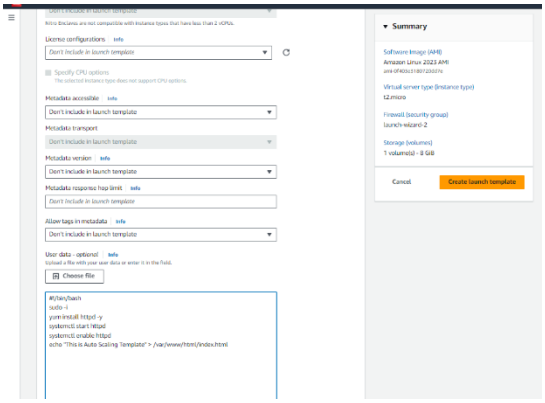



Step II : Give the name to template, select AMI and instance type then select existing group which include SSH and HTTP enable.



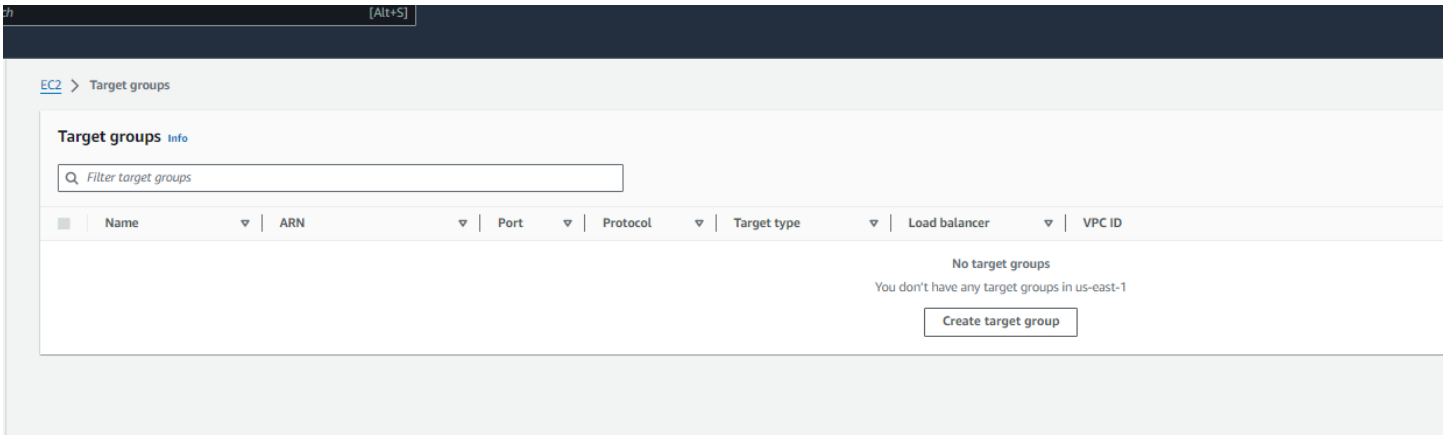



Step III : In Advanced details, write above script and then click on launch Template.

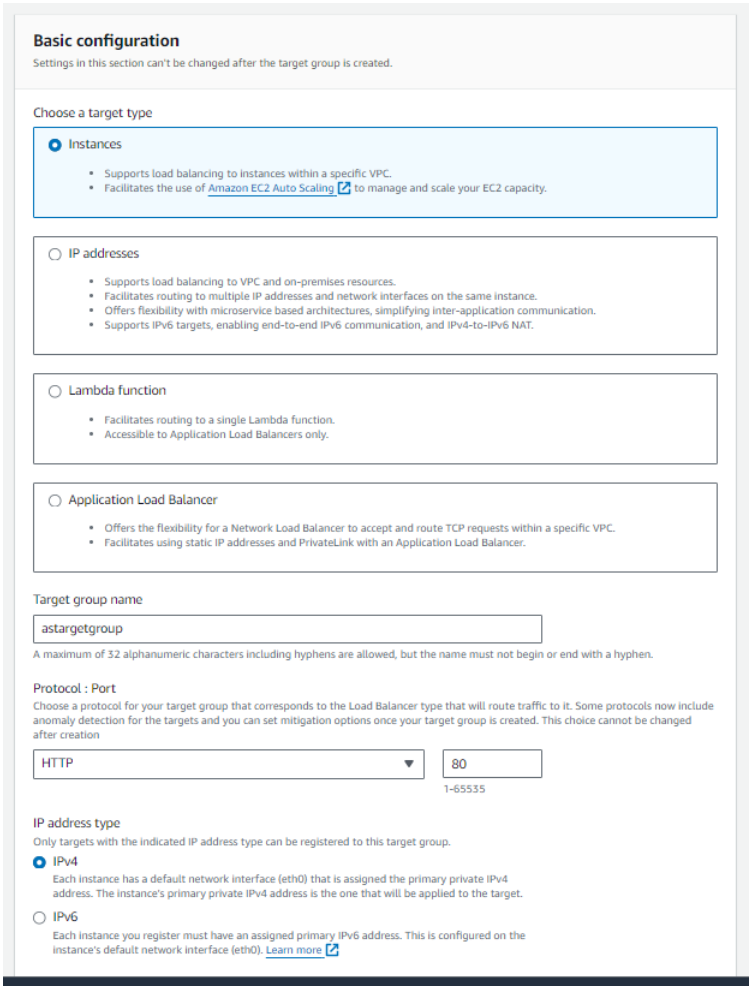


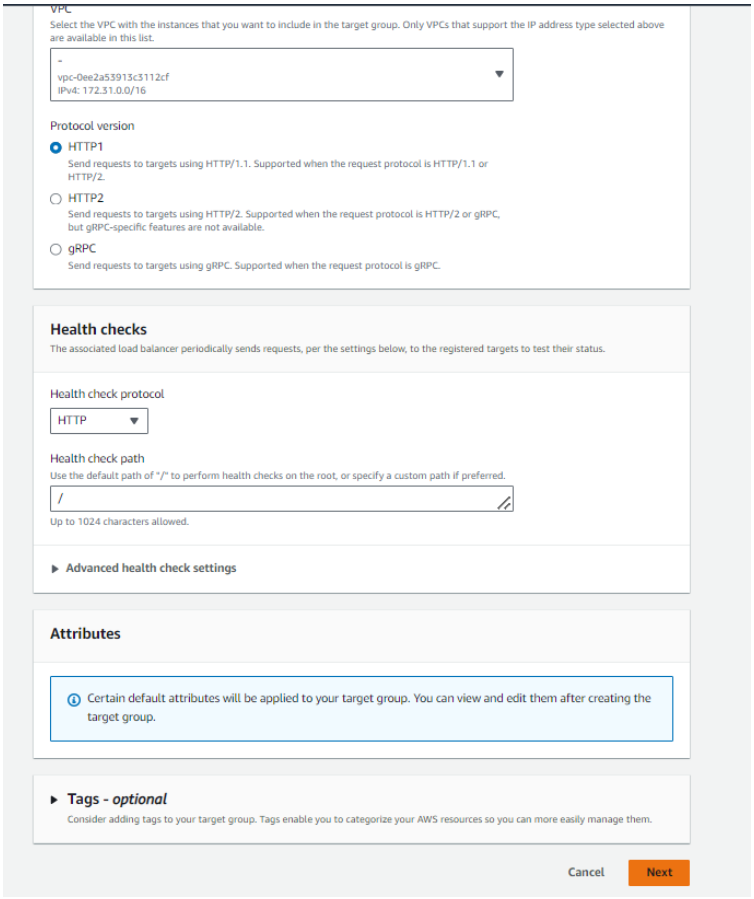

## 2) Create Target Group

Step I : Go to Ec2, In Load Balancing click on target Group then click on create target group.

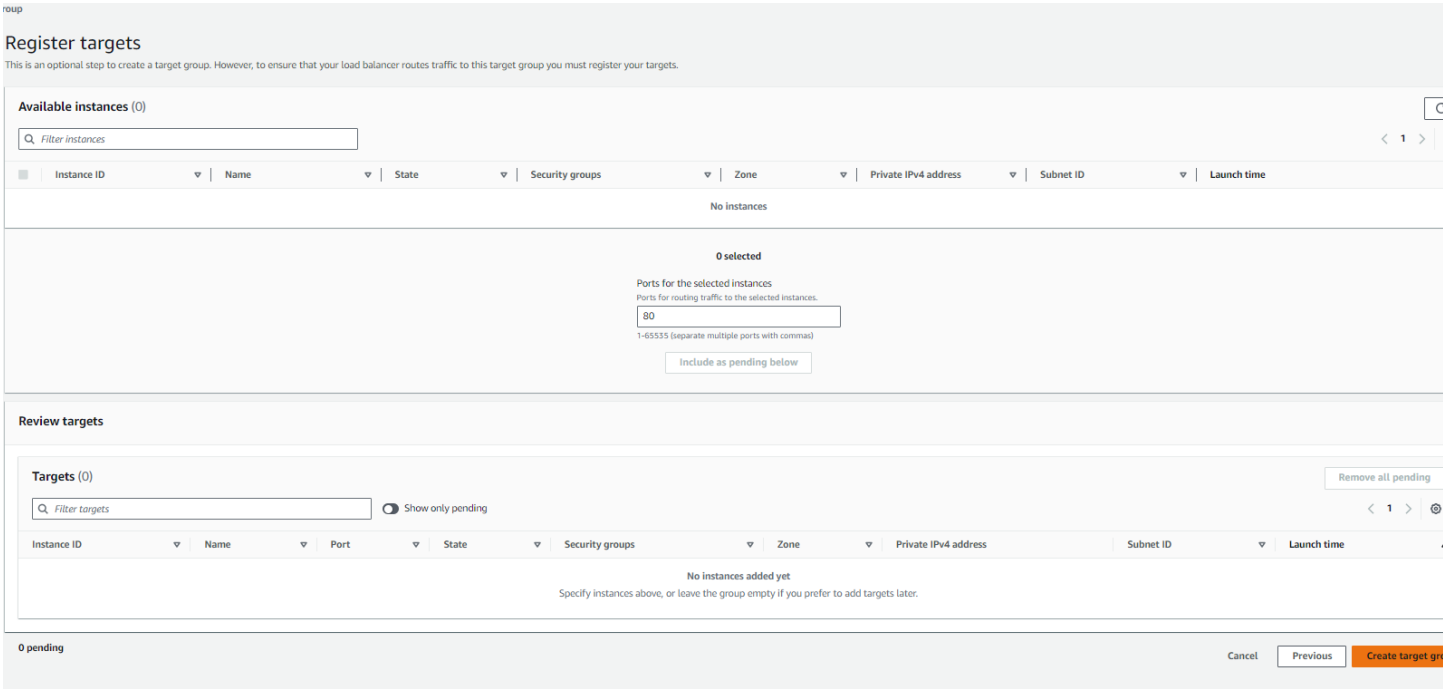



Step II : Give the name to Target group give necessary information and click on next.



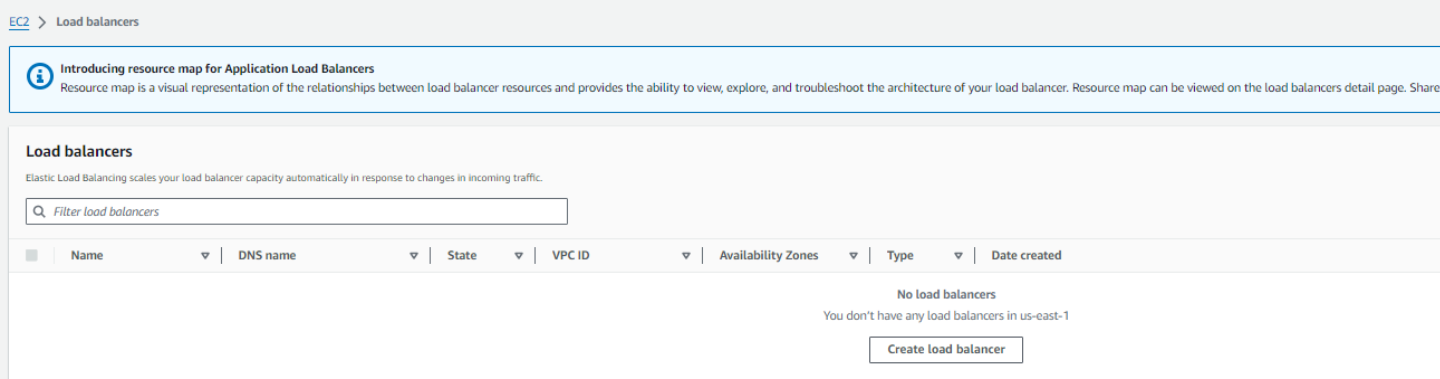



Step III : Leave all as default and click on create target group.

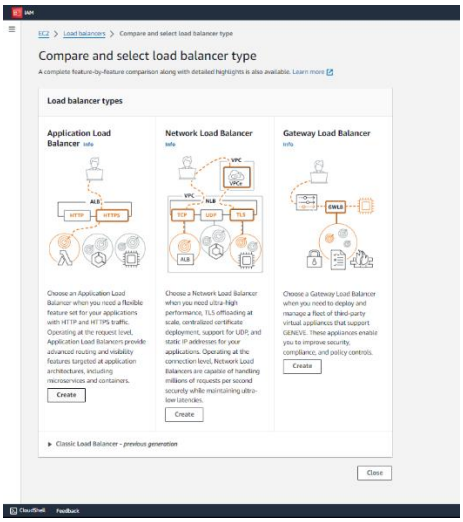


## 3) Create Load Balancer.

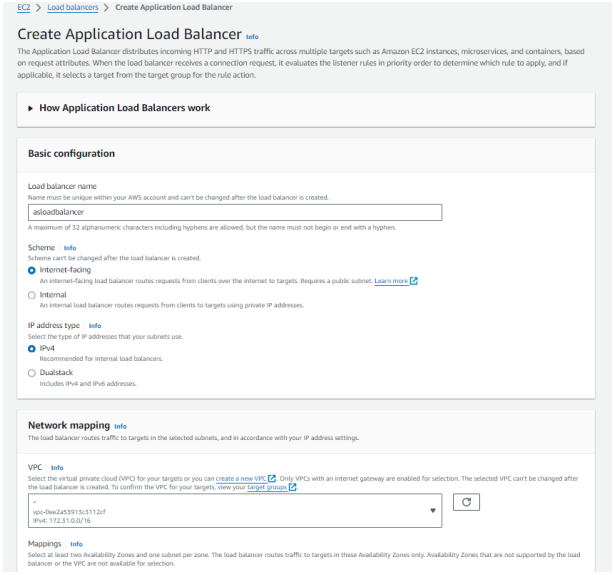
Step I : Go to EC2 ,In Load Balancing click on create Load Balancer.

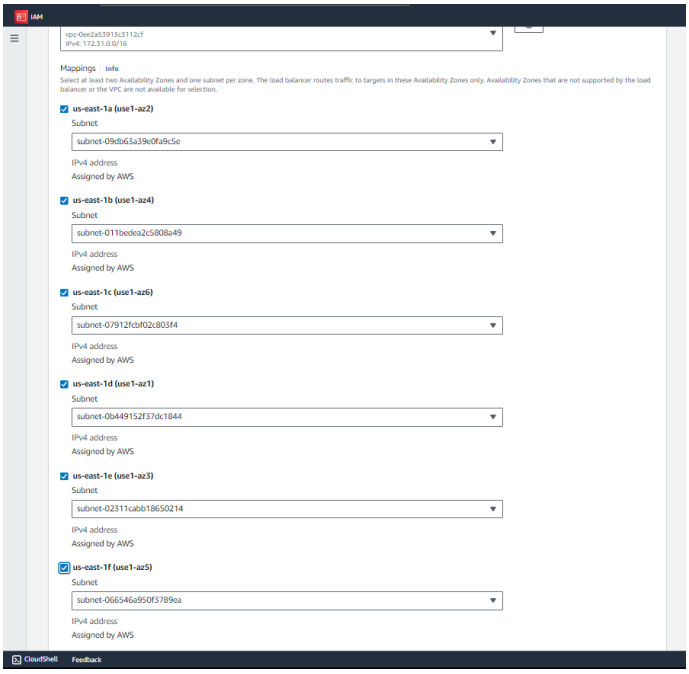


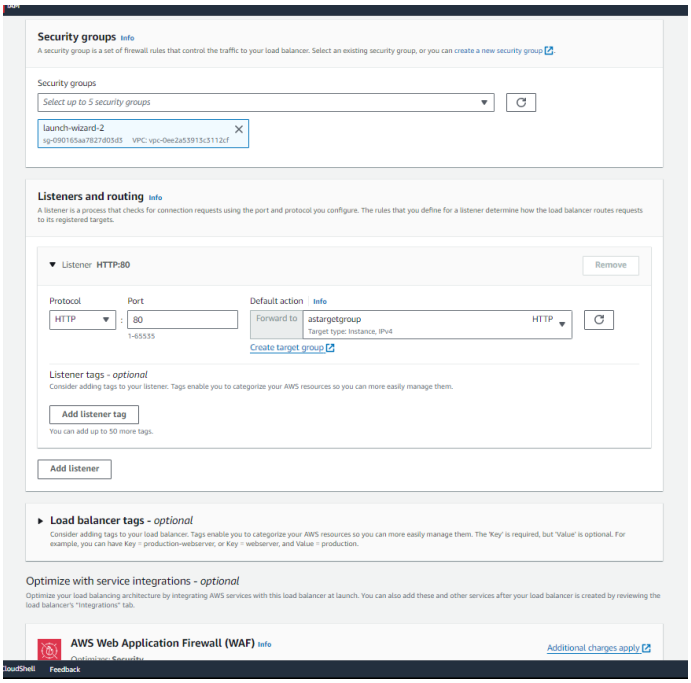

Step II : Select The application load Balancer.



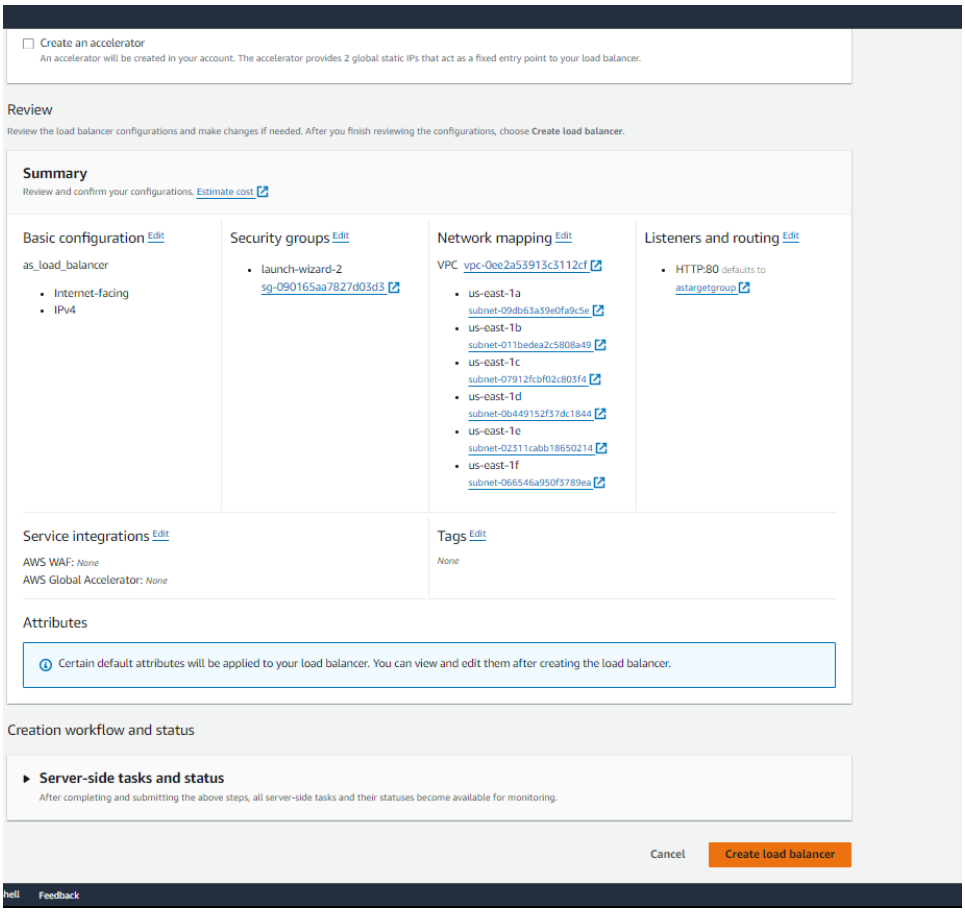

Step III : Give the name to Load Balancer, give mapping, port number and target group.


EC2 > Load balancers > Create Application Load Balancer
Create Application Load Balancer Info
The Application Load Balancer distributes incoming HTTP and HTTPS traffic across multiple targets such as Amazon EC2 instances, microservices, and containers, based on request attributes. When the load balancer receives a connection request, it evaluates the listener rules in priority order to determine which rule to apply, and if applicable, it selects a target from the target group for the rule action.
How Application Load Balancers work
Basic configuration
Load balancer name
Name must be unique within your AWS account and can't be changed after the load balancer is created.
asloadbalancer
A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.
Scheme Info
Scheme can't be changed after the load balancer is created.
Internet-facing
An internet-facing load balancer routes requests from clients over the internet to targets. Requires a public subnet. Learn more
Internal
An internal load balancer routes requests from clients to targets using private IP addresses.
IP address type Info
Select the type of IP addresses that your subnets use.
IPv4
Recommended for internal load balancers.
Dualstack
Includes IPv4 and IPv6 addresses.
Network mapping Info
The load balancer routes traffic to targets in the selected subnets, and in accordance with your IP address settings.
VPC Info
Select the virtual private cloud (VPC) for your targets or you can create a new VPC. Only VPCs with an internet gateway are enabled for selection. The selected VPC can't be changed after the load balancer is created. To confirm the VPC for your targets, view your target groups
-
vpc-0ee2a53913c3112cf
IPv4: 172.31.0.0/16
Mappings Info
Select at least two Availability Zones and one subnet per zone. The load balancer routes traffic to targets in these Availability Zones only. Availability Zones that are not supported by the load balancer or the VPC are not available for selection.


IAM
vpc-0ee2a53913c3112cf
IPv4: 172.31.0.0/16
Mappings Info
Select at least two Availability Zones and one subnet per zone. The load balancer routes traffic to targets in these Availability Zones only. Availability Zones that are not supported by the load balancer or the VPC are not available for selection.
us-east-1a (use1-az2)
Subnet
subnet-09db63a39e0fa9c5e
IPv4 address
Assigned by AWS
us-east-1b (use1-az4)
Subnet
subnet-011bedea2c5808a49
IPv4 address
Assigned by AWS
us-east-1c (use1-az6)
Subnet
subnet-07912fcbf02c803f4
IPv4 address
Assigned by AWS
us-east-1d (use1-az1)
Subnet
subnet-0b449152f37dc1844
IPv4 address
Assigned by AWS
us-east-1e (use1-az3)
Subnet
subnet-02311cabb18650214
IPv4 address
Assigned by AWS
us-east-1f (use1-az5)
Subnet
subnet-066546a950f3789ea
IPv4 address
Assigned by AWS
CloudShell
Feedback

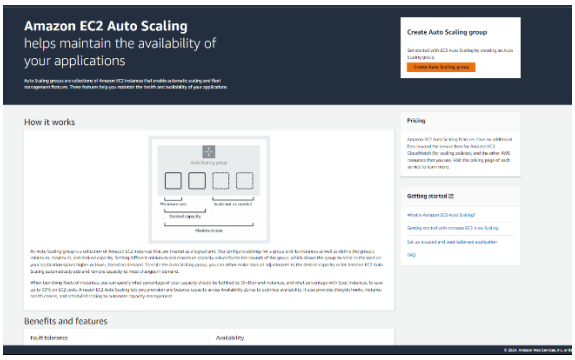



# Step IV : Click on create Load Balancer.

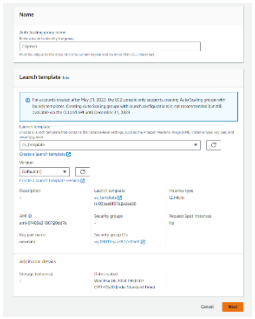


## 4) Create auto scaling group

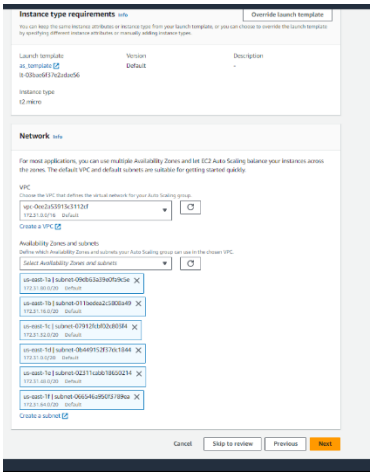
Step I : In Ec2 ,Click on auto Scaling and then create auto Scaling group.




Step II : Give the name and select launch template.click on next.

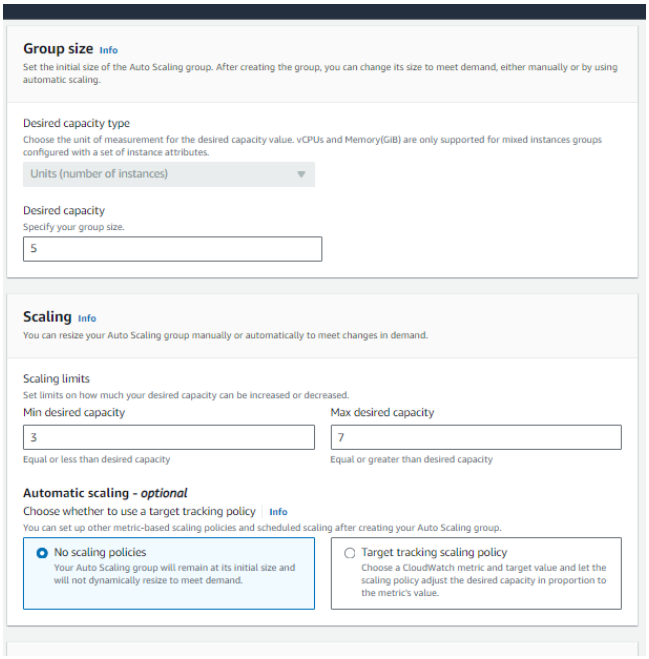


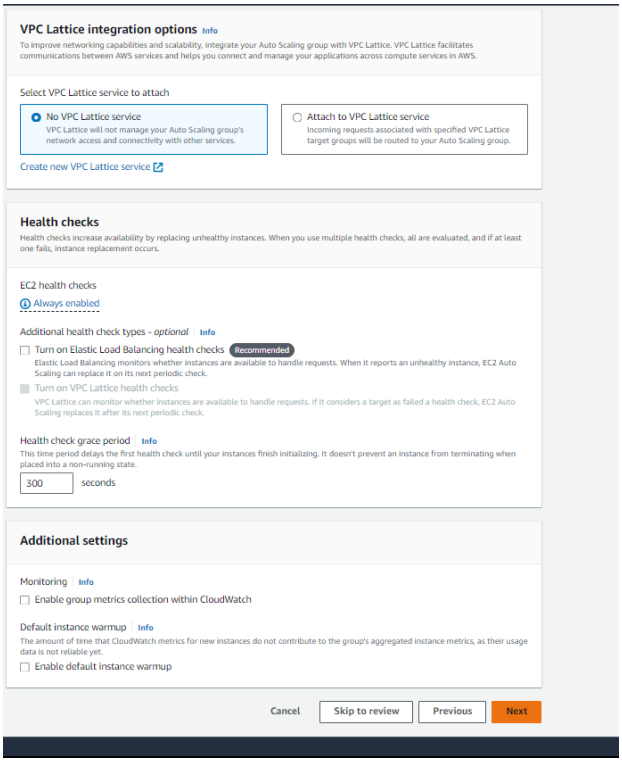

Step III : Choose Availability zones and attach it to the load balancer,and next.



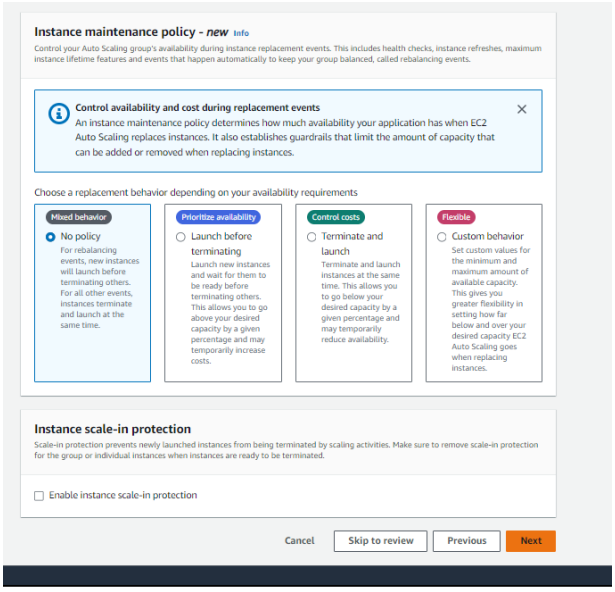
Step IV : Give the desired ,min, max size to instances also define scaling policies.

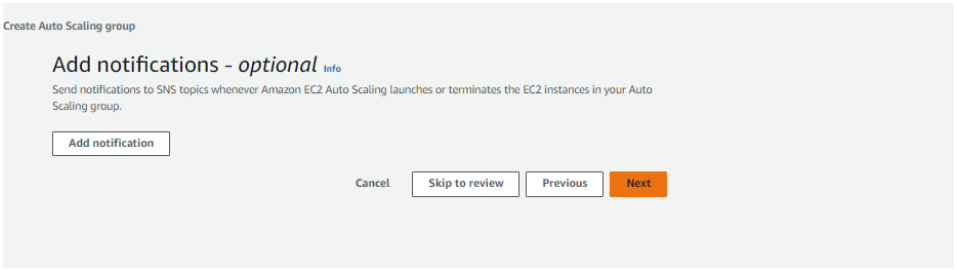
Step V : then click next.

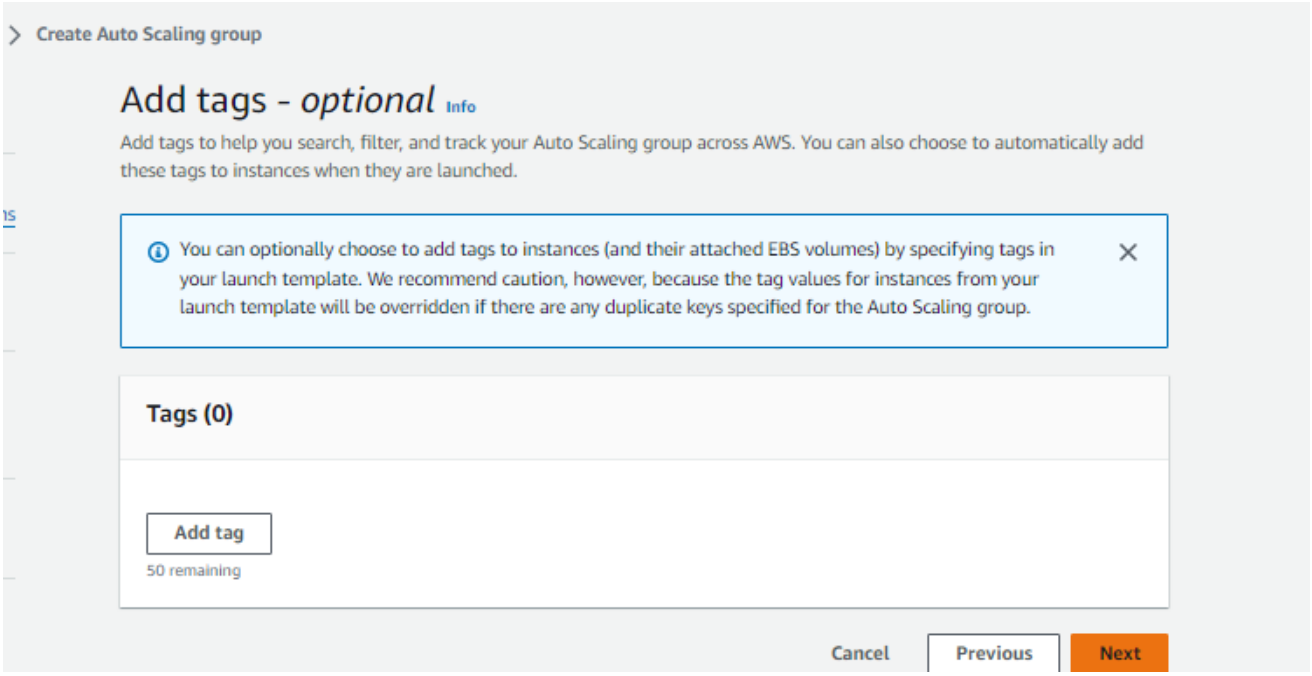


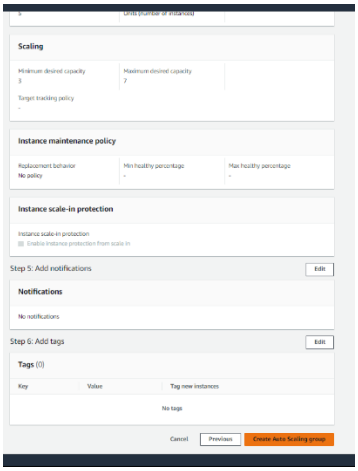
# Step VI : then click next.



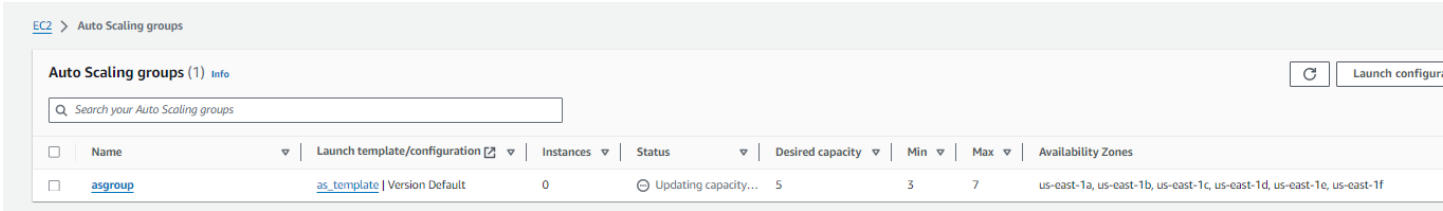



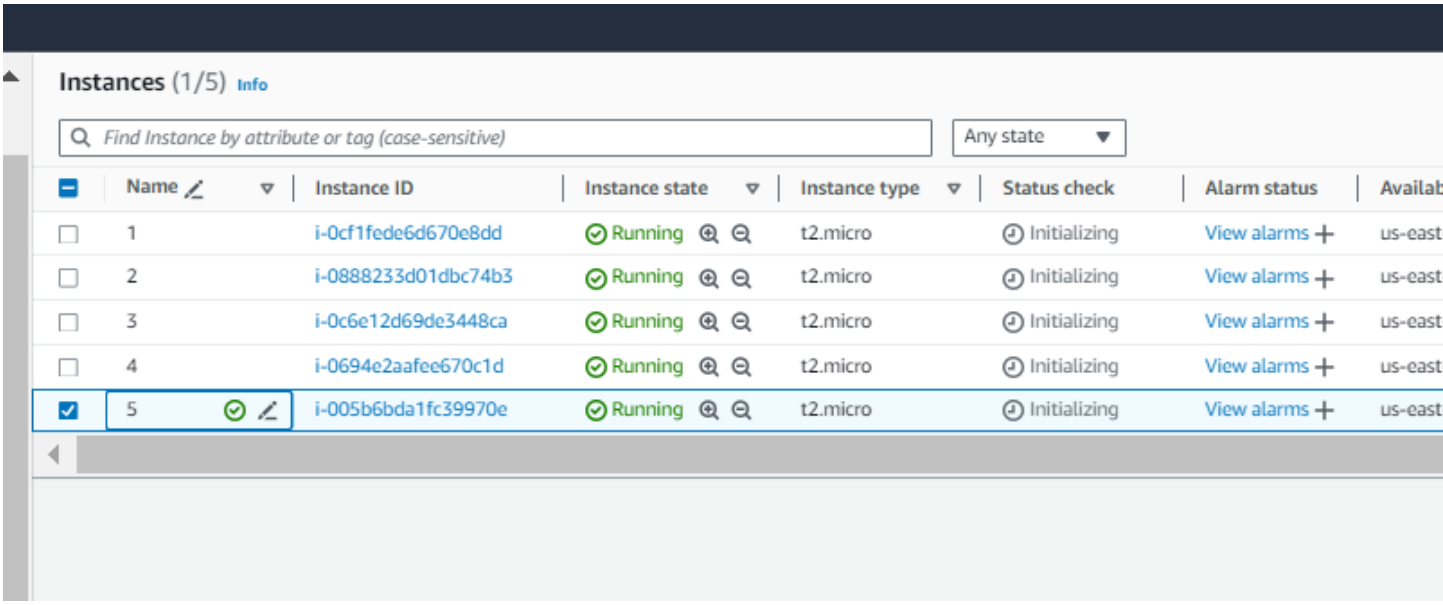
Step VII : then click on create auto scaling group.



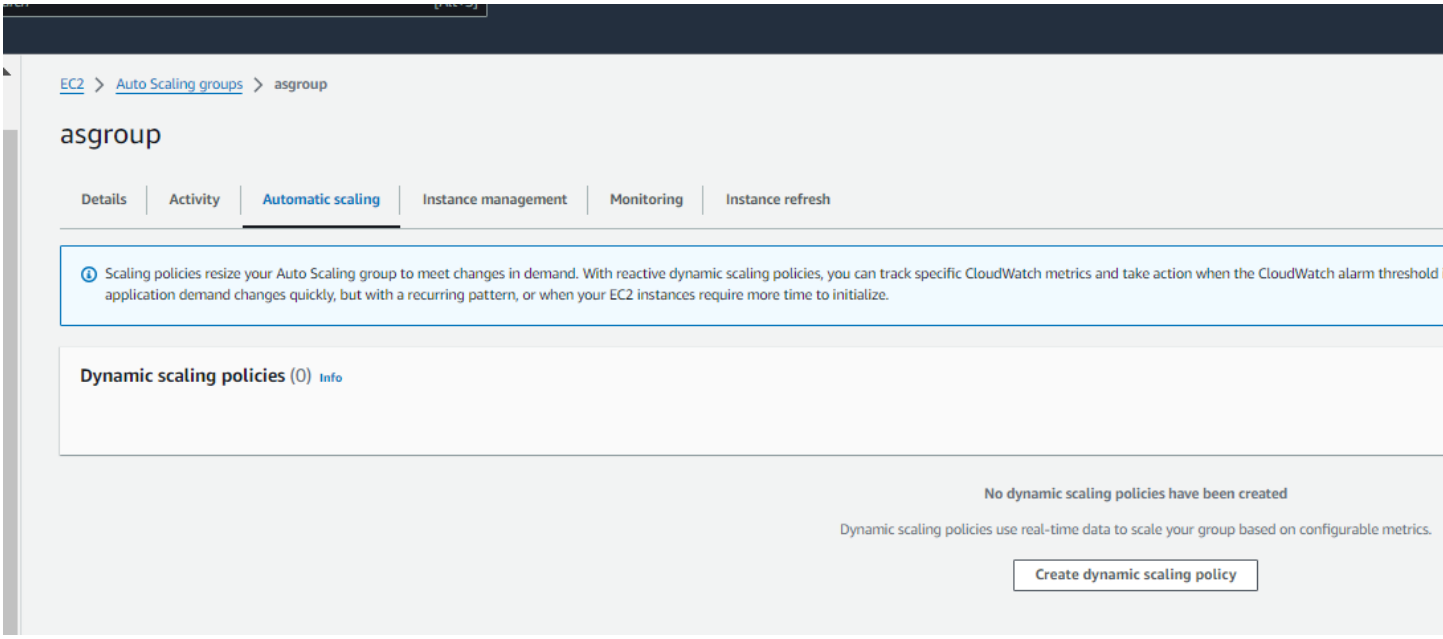

Step VIII : here,auto-scaling group is created.

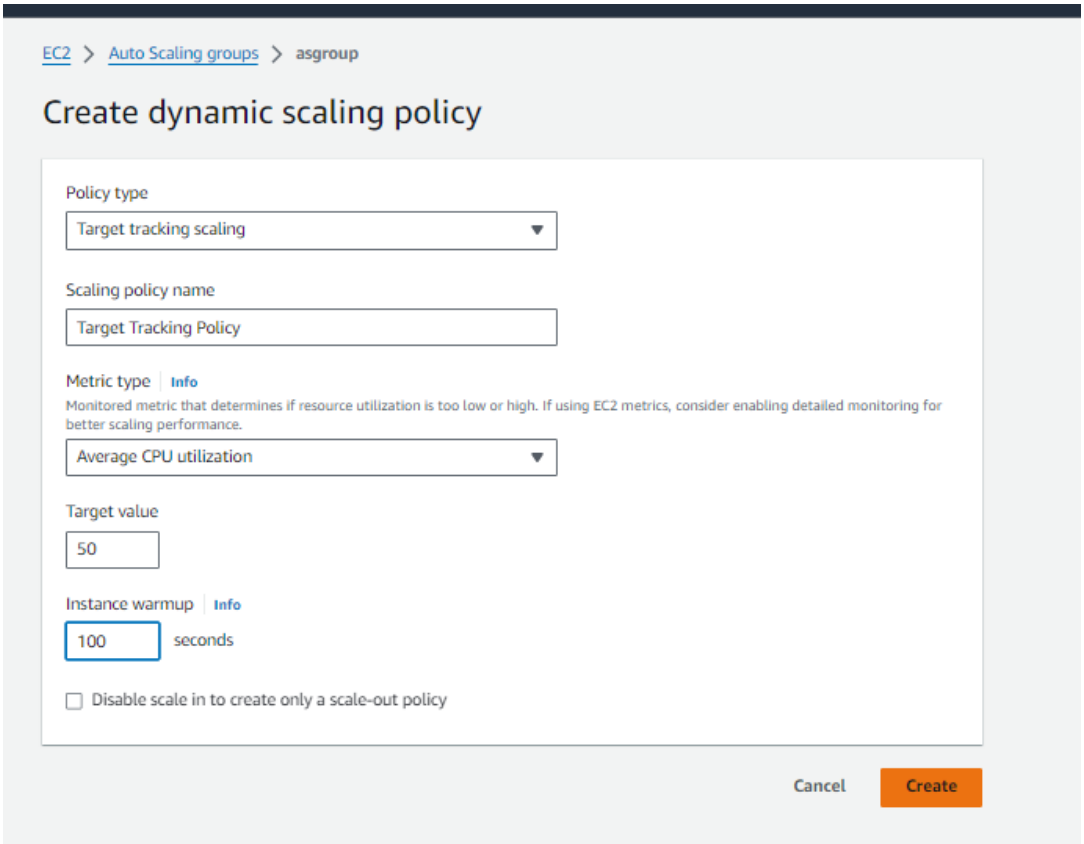


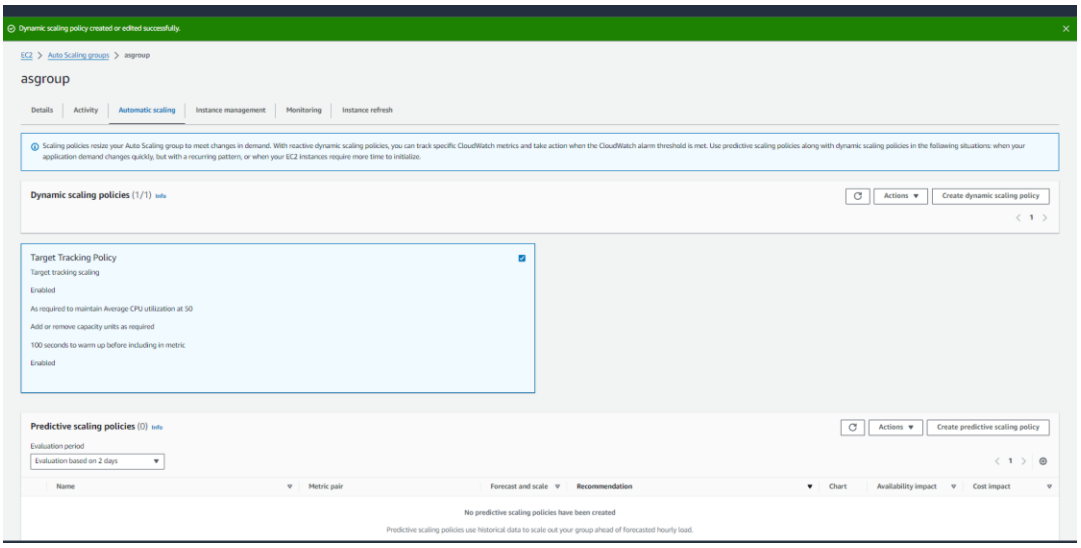

Step IX : Instances are also Launched see here ,5 instances are created here as desired capacity, name it .



Step X : click on scaling group name and go to  Automatic scaling here click on create dynamic scaling policy.






## To manage instance:

Step XI : Select any one instance copy its ip go to cmd change the directory where your key is present and then give command

i)ssh -i newdata.pem ec2-user@public_ip_of_instance it is for take remote access.

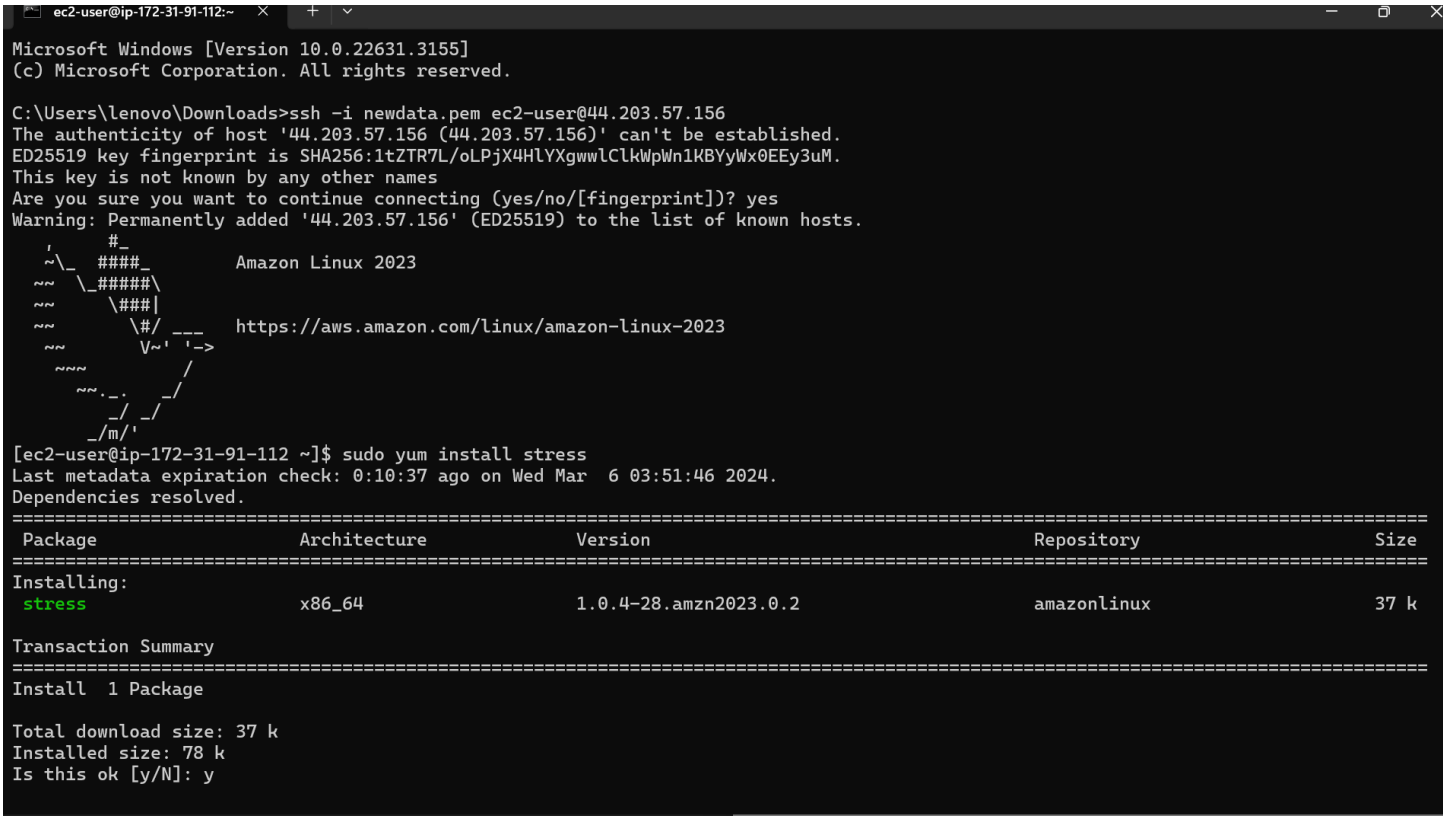
ii)install the  command using “ sudo yum install stress”

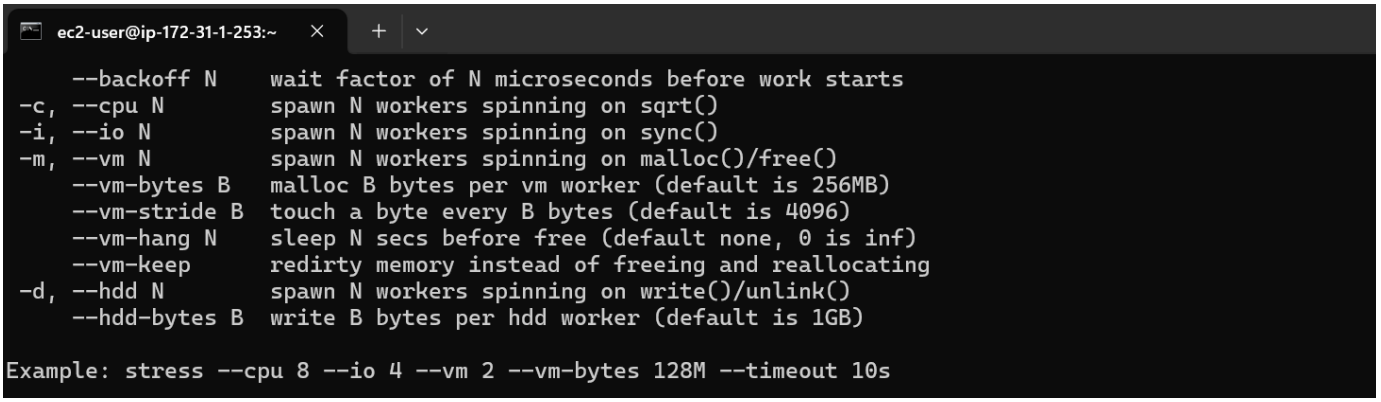
iii) after that enter command “stress –help” .

iv) copy these command change cpu 88 and give 15m and paste it for running background give “&” at the end.

v)then give “top” to check Load Average.

Follow this step to all instances to give the stress to instances and increase the load of CPU.



```
[ec2-user@ip-172-31-1-253 ~]$ stress --cpu 87 --io 4 --vm 2 --vm-bytes 128M --timeout 10m &
[1] 26620
[ec2-user@ip-172-31-1-253 ~]$ stress: info: [26620] dispatching hogs: 87 cpu, 4 io, 2 vm, 0 hdd
stress --cpu 87 --io 4 --vm 2 stress --cpu 87 --io 4 --vm 2 --vm-bytes 128M --timeout 10m &^C
[ec2-user@ip-172-31-1-253 ~]$ stress --cpu 87 --io 4 --vm 2 --vm-bytes 128M --timeout 10m &
[2] 26760
[ec2-user@ip-172-31-1-253 ~]$ stress: info: [26760] dispatching hogs: 87 cpu, 4 io, 2 vm, 0 hdd
stress --cpu 87 --io 4 --vm 2 ^C
[ec2-user@ip-172-31-1-253 ~]$ stress --cpu 87 --io 4 --vm 2 --vm-bytes 128M --timeout 10m &
[3] 26856
[ec2-user@ip-172-31-1-253 ~]$ stress: info: [26856] dispatching hogs: 87 cpu, 4 io, 2 vm, 0 hdd

[ec2-user@ip-172-31-1-253 ~]$ top
```

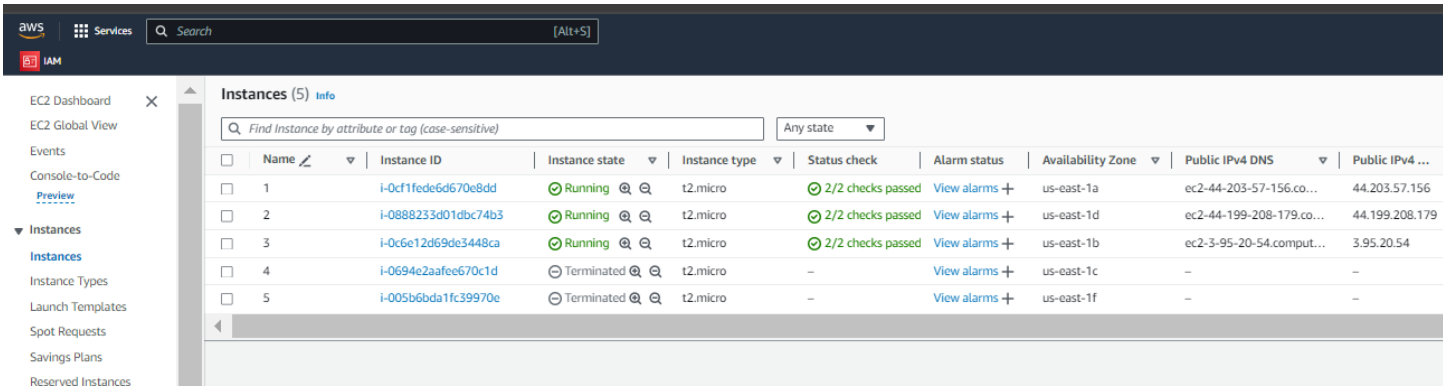
ec2-user@ip-172-31-1-253:~

```
top - 04:11:22 up 20 min,  1 user,  load average: 192.39, 66.97, 24.19
Tasks: 390 total, 270 running, 120 sleeping,   0 stopped,   0 zombie
%Cpu(s): 97.2 us,  2.8 sy,  0.0 ni,  0.0 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :    949.6 total,     90.9 free,    785.6 used,     73.1 buff/cache
MiB Swap:      0.0 total,      0.0 free,      0.0 used.     56.3 avail Mem
```

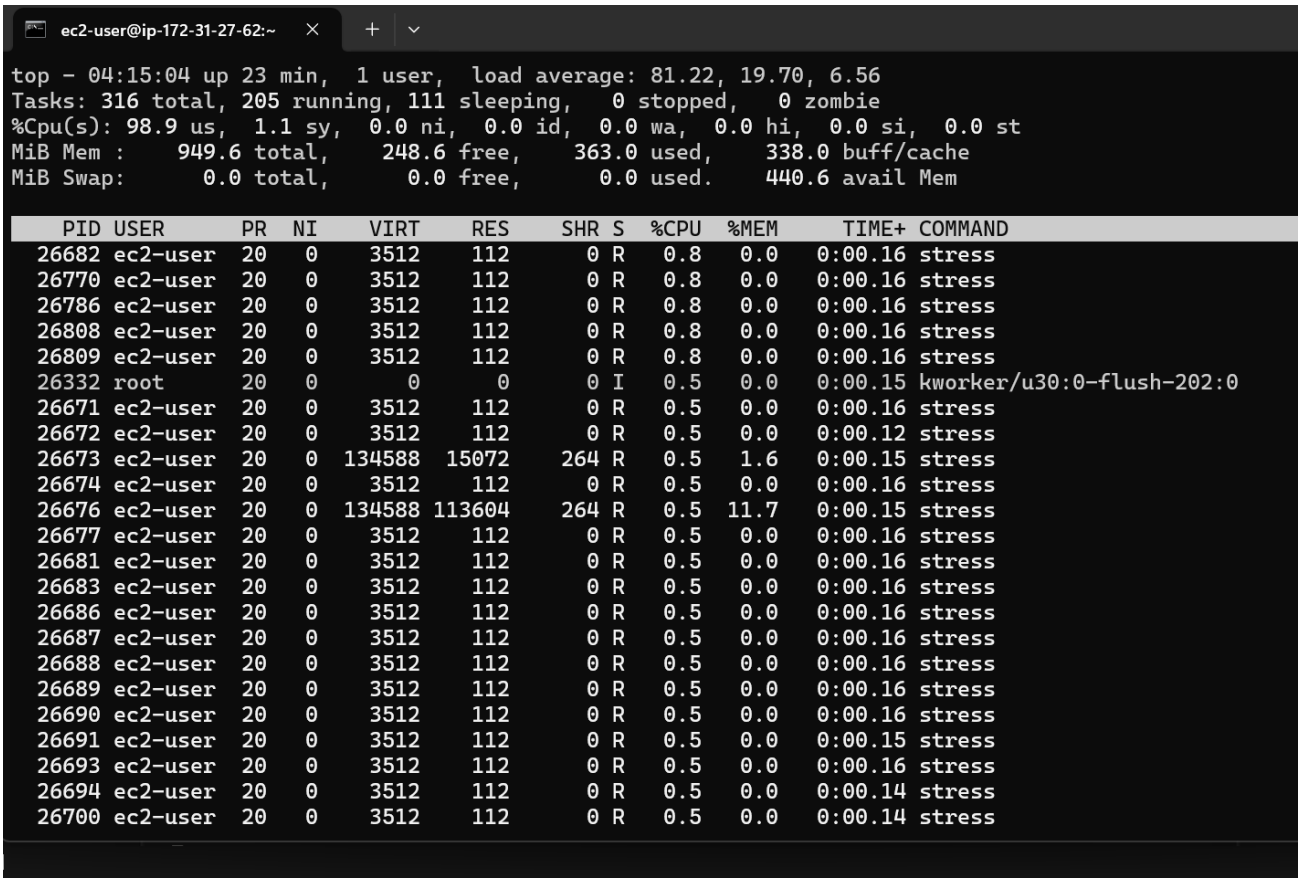
| PID | USER | PR | NI | VIRT | RES | SHR | S | %CPU | %MEM | TIME+ | COMMAND |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 26825 | ec2-user | 20 | 0 | 3512 | 112 | 0 | R | 0.7 | 0.0 | 0:00.43 | stress |
| 26862 | ec2-user | 20 | 0 | 134588 | 93296 | 264 | R | 0.7 | 9.6 | 0:00.12 | stress |
| 26623 | ec2-user | 20 | 0 | 134588 | 117584 | 264 | R | 0.5 | 12.1 | 0:00.86 | stress |
| 26624 | ec2-user | 20 | 0 | 3512 | 108 | 0 | R | 0.5 | 0.0 | 0:00.86 | stress |
| 26626 | ec2-user | 20 | 0 | 134588 | 96992 | 264 | R | 0.5 | 10.0 | 0:00.86 | stress |
| 26627 | ec2-user | 20 | 0 | 3512 | 108 | 0 | R | 0.5 | 0.0 | 0:00.86 | stress |
| 26629 | ec2-user | 20 | 0 | 3512 | 108 | 0 | R | 0.5 | 0.0 | 0:00.85 | stress |
| 26631 | ec2-user | 20 | 0 | 3512 | 108 | 0 | R | 0.5 | 0.0 | 0:00.86 | stress |
| 26632 | ec2-user | 20 | 0 | 3512 | 108 | 0 | R | 0.5 | 0.0 | 0:00.86 | stress |
| 26636 | ec2-user | 20 | 0 | 3512 | 108 | 0 | R | 0.5 | 0.0 | 0:00.86 | stress |
| 26638 | ec2-user | 20 | 0 | 3512 | 108 | 0 | R | 0.5 | 0.0 | 0:00.86 | stress |
| 26639 | ec2-user | 20 | 0 | 3512 | 108 | 0 | R | 0.5 | 0.0 | 0:00.86 | stress |
| 26640 | ec2-user | 20 | 0 | 3512 | 108 | 0 | R | 0.5 | 0.0 | 0:00.86 | stress |
| 26641 | ec2-user | 20 | 0 | 3512 | 108 | 0 | R | 0.5 | 0.0 | 0:00.86 | stress |
| 26643 | ec2-user | 20 | 0 | 3512 | 108 | 0 | R | 0.5 | 0.0 | 0:00.86 | stress |
| 26644 | ec2-user | 20 | 0 | 3512 | 108 | 0 | R | 0.5 | 0.0 | 0:00.86 | stress |
| 26646 | ec2-user | 20 | 0 | 3512 | 108 | 0 | R | 0.5 | 0.0 | 0:00.86 | stress |
| 26647 | ec2-user | 20 | 0 | 3512 | 108 | 0 | R | 0.5 | 0.0 | 0:00.86 | stress |
| 26649 | ec2-user | 20 | 0 | 3512 | 108 | 0 | R | 0.5 | 0.0 | 0:00.86 | stress |
| 26650 | ec2-user | 20 | 0 | 3512 | 108 | 0 | R | 0.5 | 0.0 | 0:00.86 | stress |
| 26652 | ec2-user | 20 | 0 | 3512 | 108 | 0 | R | 0.5 | 0.0 | 0:00.86 | stress |
| 26653 | ec2-user | 20 | 0 | 3512 | 108 | 0 | R | 0.5 | 0.0 | 0:00.86 | stress |
| 26654 | ec2-user | 20 | 0 | 3512 | 108 | 0 | R | 0.5 | 0.0 | 0:00.86 | stress |

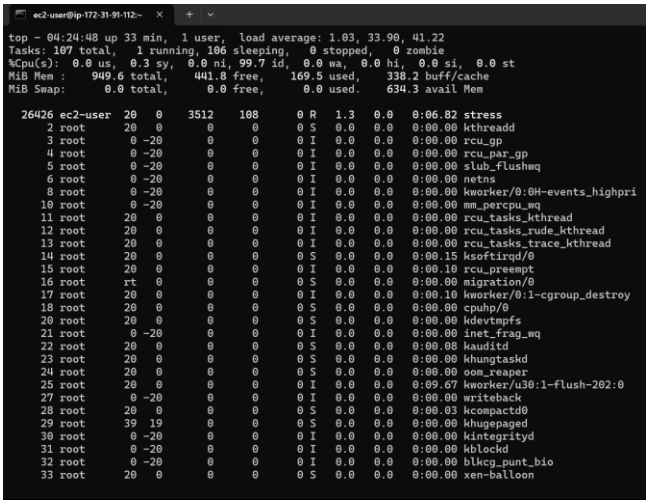
Daily Alina.pem 2/29/2024 9:50 PM PEM File

Step XII : Here, load is minimum that why here only minimum capacity of instances running.

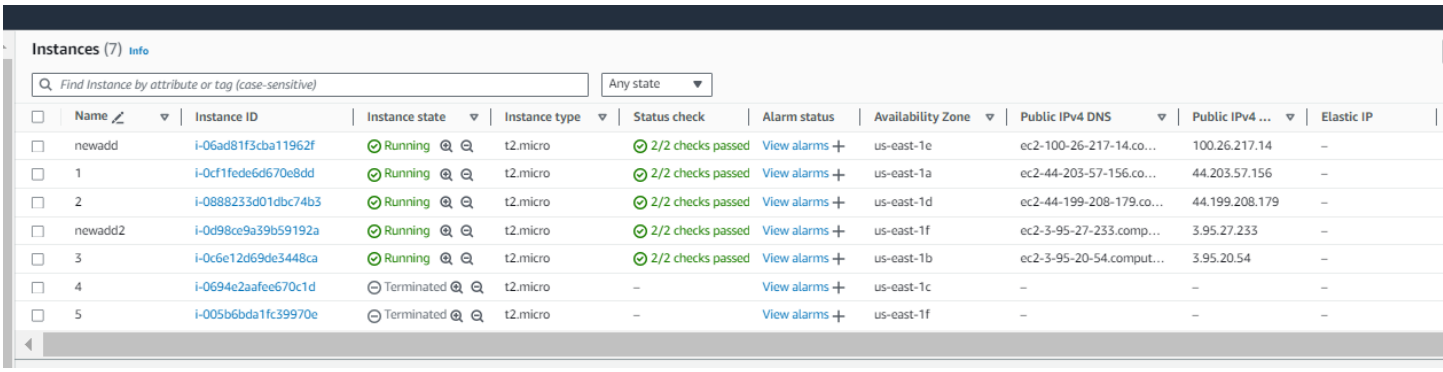



Step XIII : we add maximum load on it using above command for all instnces






Step XIV : here see to load increased that's why maximum capacity of instances are add to distribute load.




Step XV : After doing that, kill the stress give command "killall stress".